КИЇВСКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

РОЗРОБКА СИСТЕМИ

АВТОМАТИЗАЦІЇ CALL-ЦЕНТРУ, ТЕЛЕФОННИХ РОЗМОВ

ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ AMI

Виконав студент 4-го курсу
Артюх Владислав Павлович

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Кузенко Володимир Федорович

(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент

(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри теорії та технології програмування
«____» _____ 201_ р.,
протокол № ___
Завідувач кафедри
Нікітченко М. С

Київ-2021

РЕФЕРАТ

Обсяг роботи 41 сторінка, 31 ілюстрація, 1 таблиця, 6 джерел.

CALL-ЦЕНТРИ, АВТОМАТИЗАЦІЯ РОБОТИ, PHP, AMI

Об’єктом є розробка веб-сервісу автоматизації роботи установ, що спеціалізуються на прийомі та обробці дзвінків.

Метою роботи є створення системи автоматизованого додатку що ґрунтується на технології AMI.

Методи розроблення: аналіз існуючих проблем, проектування майбутньої архітектури програми, покрокова розробка з розбиттям на під задачі.

Інструменти розроблення: Для розробки даної програмної реалізації було обрано мову PHP. Під час розробки у якості інтегрованого середовища було обрано PHP Storm. Для розробки Frontend частини було використано HTML та CSS.

Результат роботи: проведено порівняльний аналіз існуючих на ринку технологій з автоматизації роботи call-центрів, виділено переваги та недоліки кожного з них, розглянуто архітектуру конструкторів розмовних скриптів. Проведено аналіз проблем, які виникають під час розробки аналогічних програмних застосунків. В результаті було спроектовано гнучкий конструктор скриптів та автоматизовано всі процеси взаємодіє операторів та клієнтів за допомогою технології AMI.

Програмний засіб можна використовувати як систему автоматизації call-центру з метою покращення якості та ефективності роботи установи.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

БД – База даних

Call-centre – установа що спеціалізується на прийомі, виконанні та опрацюванні дзвінків

Script (Скрипт) – в контексті даної роботи даний вираз використовується як сценарій розмови

API - application programming interface

FTP - File Transfer Protocol

AMI - Asterisk Manager Interface

Asterisk - вільне рішення комп'ютерної телефонії (в тому числі, VoIP) з відкритим вихідним кодом від компанії Digium

VoIP – IP телефонія, тобто телефонний зв'язок по протоколу IP

UX-дизайн - це проектування інтерфейсу на основі досліджень користувацького досвіду і поведінки

UI-дизайн - процес візуалізації прототипу, який розробили на підставі призначеного для користувача досвіду і дослідження цільової аудиторії.

Backend – клієнтська візуальна сторона сервісу призначена для взаємодії з програмно-апаратною стороною

Frontend – програмно-апаратна частина сервісу

ВСТУП

Оцінка сучасного стану об’єкта розробки

Сучасна веб-система автоматизації розмов, повинна враховувати всі види послуг що може надавати call-центр та потреби користувачів, і при цьому залишатися ефективною та легкою в користуванні.

Актуальність роботи та підстави її виконання

На сьогодні телекомунікаційні технології відіграють невід’ємну роль у нашому світі, жодна сучасна галузь не може існувати за відсутності якісного та швидкого зв’язку. Завдяки швидкому розвитку даних технологій, якісні та швидкі міжнародні та міжрегіональні дзвінки, а саме головне за низькою ціною, стали буденністю.

Разом з популяризацією телекомунікації з’явилася потреба в розробці систем автоматизації, моніторингу та покращення зручності користування. Посередником у більшості телефонних дзвінків у сфері послуг - являється call-центр. Звідси виникає проблема оптимізації робити даної установи, зокрема в побудові чіткого алгоритму розмови оператора та клієнта.

Мета й завдання роботи

Метою даної робити є розробка веб-системи, яка допоможе автоматизувати і значно спростити роботу працівників call-центру, в особливості під час спілкування з клієнтами. Для досягнення даної мети поставлено відповідні завдання:

- Зробити детальний аналіз роботи call-центру, а саме структури послуг що надаються;
- На основі отриманих даних спроектувати веб-застосунок та його БД;

- Розробити ПП, який буде повністю відповідати спроектованій системі;
  - реалізувати ефективний конструктор алгоритму розмови
  - розробити систему автоматичної обробки дзвінків
- Забезпечити надійність користування ПП, та цілісність даних.

Об'єкт, методи й засоби розроблення

Об’єктом розробки став веб-застосунок «CallBase», що дозволить оператору здійснювати розмову з клієнтом, по раніше заготовленому сценарію, а також реалізує автоматичне отримання і подальшу обробку дзвінків, що обумовить кращу швидкість та якість обробки запиту, наданого клієнтом.

Для розробки даної системи був проведений аналіз потреб користувачів, та всіх видів послуг call-центру.

Можливі сфери застосування

Проект може бути використаним будь-якою компанією що надає телекомунікаційні послуги, або ж працює в сфері телефонних продажів.

1. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА ЇХ ОПИС

1.1 Мови розробки

Є три головні етапи розробки, а саме frontend, backend та взаємодія з Asterisk. Для реалізації frontend розробки було використано HTML версії HTML5 та CSS версії CSS3. Дані мови програмування та розмітки було вибрано у зв'язку з тим що вони є пануючими на ринку веб-ресурсів, і вважаються основними при розробці frontend частини.

Backend було написано мовою PHP [2] разом із технологією AMI, яка й використовувалась для підключення та взаємодії з Asterisk, а також JavaScript. PHP є однією із найбільш поширених мов програмування при розробці у веб сегменті, вона вважається традиційною мовою, її найбільші переваги це простота та ефективність.

1.2 Основне середовище розробки

Головним середовищем розробки було використано PHP Storm, найпопулярніше середовище для роботи с такою мовою програмування як PHP, надає низку можливостей які стали ключовим фактором у виборі середовища:

- Підтримка БД та SQL;
- Можливість віддаленого розгортання та автоматична синхронізація використовуючи основні протоколи (FTP, SFTP та інші);
- Історія локального рівня;

- Система що відстежує помилки в коді;
- Можливе поєднання клавіш для того щоб підвищити ефективність і швидкість роботи.

Також в даному середовищі передбачена робота з JavaScript, що також використовувався в процесі розробки, HTML та CSS.

1.3 Середовище роботи з БД

Для роботи з БД було обрано phpmyadmin — веб-додаток з відкритим кодом на мові PHP із графічним web-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Зображених в браузерах і які все ще правильно відображаються як веб-сторінки.

1.4 Допоміжне середовище розробки

В ролі допоміжного середовища розробки виступав Sublime Text — швидкий багатомовний текстовий редактор. Підтримує плагіни, розроблені за допомогою мови програмування Python. Sublime Text не є вільним чи відкритим програмним забезпеченням, але деякі його плагіни розповсюджуються з вільною ліцензією, розробляються і підтримуються спільнотою розробників. Використовувався мною лише для написання фалів мовами HTML та CSS.

1.5 AMI Астеріск

AMI [1], [4] одна з ключових технологій мого проекту – це програмний продукт для керування системою Астеріск [3] із зовнішніх програм. Завдяки даній технології зовнішні програми мають змогу підключитися до системи Астеріск за допомогою TCP протоколу та ініціювати виконання та зчитувати результат роботи наданих команд, також технологія дозволяє моніторити всі події що відбуваються в системі в реальному часі.

Перевага AMI в тому що він не потребує прямого доступу до сервера на якому знаходиться Астеріск. При правильному використанні даної технології можливе створення ПП що забезпечить продуктивну і швидку роботу процесів call-центру, завдяки зчитуванні всіх подій Астеріск в реальному часі.

Взаємодія з робочим сервером відбувається за допомогою простого по-рядкового протоколу, а саме двох рядків:

- Ключове слово (key), яке являється унікальна і позначає вид і характер переданої інформації;
- Значення переданого параметра (value).

Конструкції виду “ key:value “ формують так званий “ пакет ”. Перед тим як посилати команди в сервер Астеріск, необхідно виконати сесію з'єднання клієнта з сервером Астеріск. Пакети можуть бути передані в будь-якій послідовності і в будь-який час після проходження процедури автентифікації; Перший рядок пакета повинна містити один з таких ключів: «Action» (єдиний варіант при відправки клієнтом), і ключі «Event» (Подія) і «Response» (Відповідь) (повинні бути відправлені від Астеріск до клієнта).

Порядок рядків в пакеті не має значення, ви можете використовувати будь-яку мову програмування, яким можна формувати пакети на стороні клієнта. Але потрібно пам’ятати, що перед використанням даної технології Астеріск повинен бути налаштований до роботи з даною технологією.

1.6 Середовище взаємодії з FTP сервером

Для взаємодії з сервером було використано технологію FileZilla. Filezilla – це програмний продукт у вільному доступі, що являє собою багатоплатформний FTP клієнт з відкритим кодом.

2. ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ ТА ЇХ АНАЛІЗ

2.1 NEURON System

На ринку веб-систем для автоматизації телефонних розмов існує велика кількість програмних продуктів. Перед проектуванням власної системи було оглянуто велику частину даних продуктів, та обрано один, який на мою думку являється одним з найкращих представників в даному сегменті.

NEURON System – система автоматизації телефонних продажів. NEURON System окрім хорошої і високо деталізованої статистики включає в себе зручний конструктор алгоритму розмови.

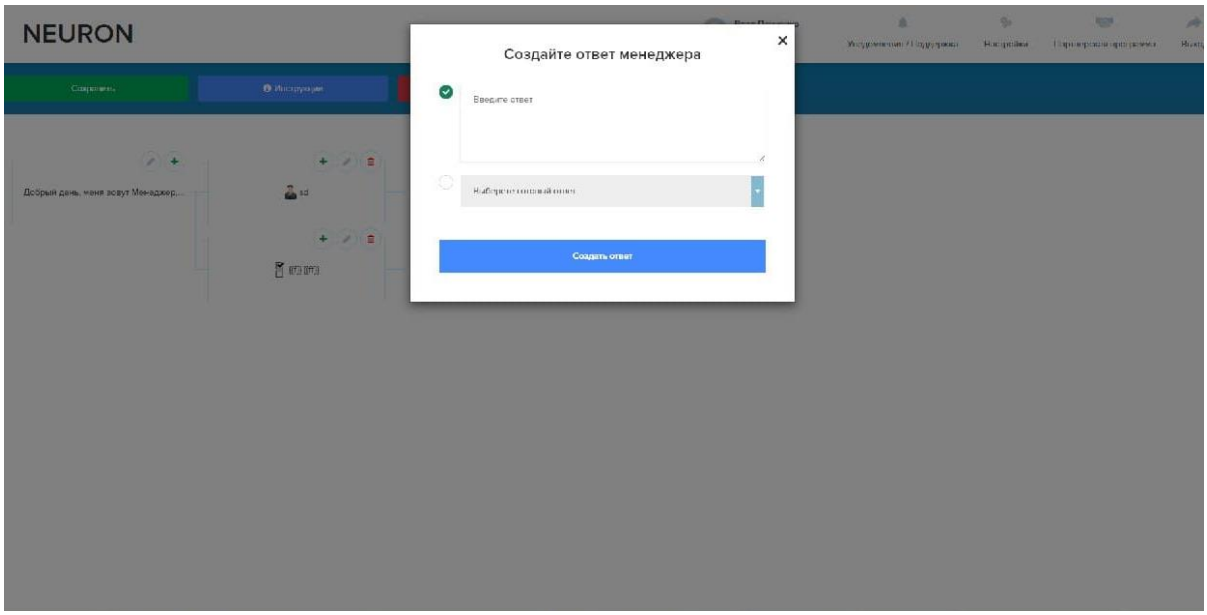


Рис 2.1 Створення блоку скрипта

На Рис 2.1 можна побачити форму додавання одного з блоків алгоритму розмови, а саме блоку що відповідає за відповідь менеджера на попереднє питання

клієнта. Алгоритм розмови являє собою чітку послідовність реплік оператора або менеджера у відповідь на певні запитання клієнта, які також в загальній формі передбачаються алгоритмом.

Не зважаючи на зручність конструктора алгоритмів NEURON даної системи є недолік в тому, що даний конструктор дозволяє будувати однотипні структури, в чіткій послідовності менеджера та клієнта, проте на практиці можна зрозуміти, що дана структура не є зовсім вірною, іноді менеджер повинен задати декілька запитань, або ж сказати декілька фраз, різних за своєю структурою і спрямуванням.

Функціонал даної системи дозволяє будувати лише “Вихідний скрипт”, по при те що існують ще два види скриптів: “Вхідний та “Пропозиція”.

Кожен з цих видів є унікальним, і суттєво відрізняється один від одного, тому реалізація функціоналу кожного з видів вимагає особливого підходу та окремого проектування, детальніше можна ознайомитися з видами скриптів розмови в таблиці.

Таблиця 1.1 Види розмовних скритів

Вихідний	Вхідний	Пропозиція
Ініціатором дзвінка є оператор(менеджер), він же і керує розмовою, тобто робить все щоб клієнт притримувався алгоритму	Ініціатором дзвінка є клієнт, зазвичай все зводиться до заповнення анкети	Даний вид скрипта можна ще назвати розсилко. Відправляє до цього заготовлений шаблон клієнту на пошту
Складна, розгалужена структура, з великою кількістю тексту та інших значень	Проста структура, яка являє собою послідовність полів яку потрібно заповнити згідно даних які надає клієнт, тобто анкета.	Складається із двох частин, створення листа та процесу відправлення

3. ПРОЕКТУВАННЯ, РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ТА ОГЛЯД РОЗРОБЛЕНОЇ СИСТЕМИ

За основу в проектуванні був взятий раніше згаданий веб-сервіс NEURON, оскільки модель даної система не задовольняє всі потреби call-центру [5], вона зазнала певних змін.

1.1 Проектування системи

- Спроектований застосунок повинна містити наступні компоненти:
- Авторизація;
- Персональна сторінка адміністратора;
- Головний стіл;
- Таблиця працівників;
- Персональна сторінка працівника;
- Конструктор скриптів;
- Статистика дзвінків;
- Таблиця тарифів;
- Налаштування;
- Статус активності.

Розроблена система містить два види користувачів: адміністратор та оператор. Адміністратор виступає в ролі менеджера call-центру, тобто в його можливості входить доступ до всіх даних та можливість їх редагування *(див рис 3.1)*.

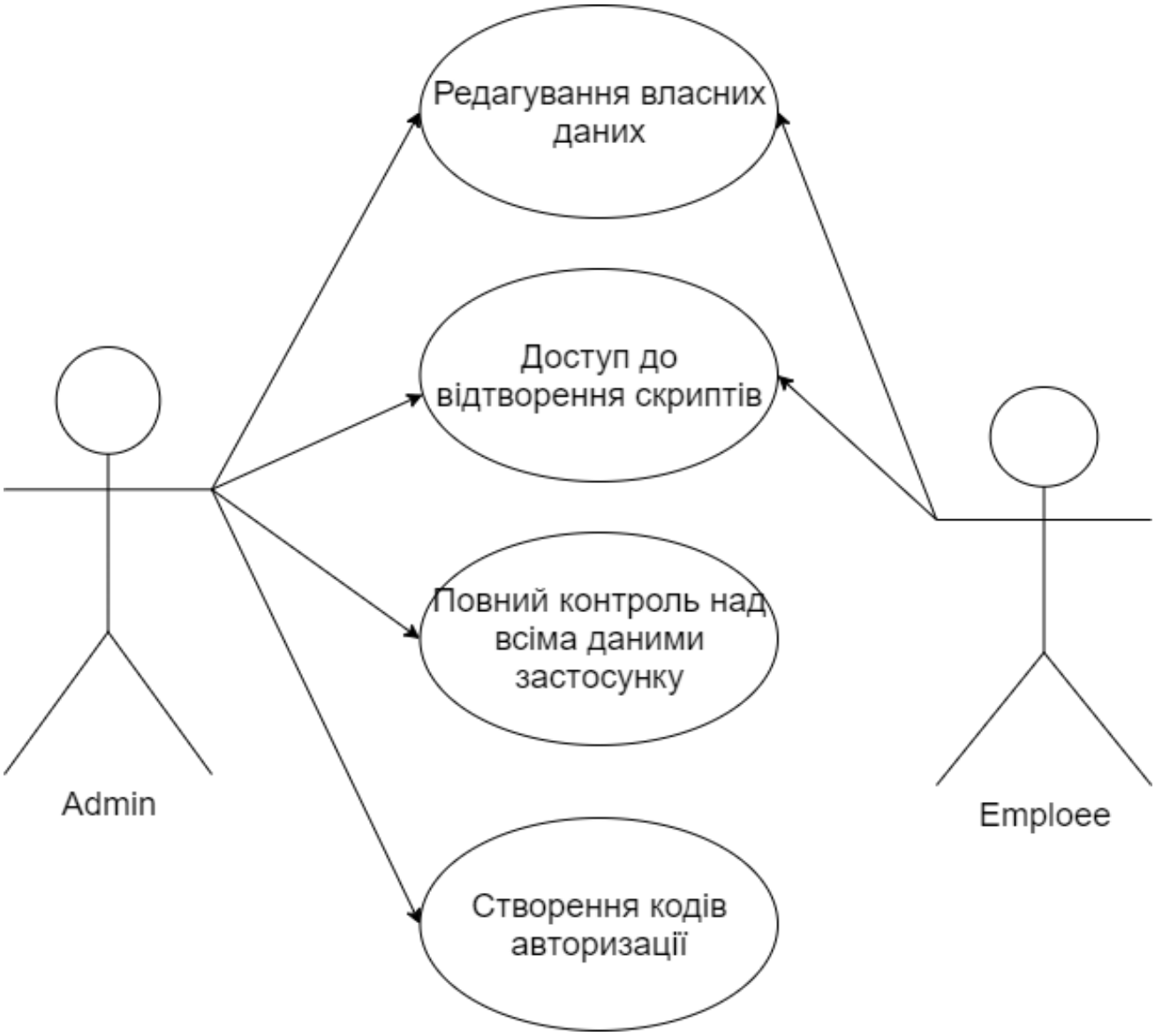


Рис 3.1 Види користувачів та їх можливості

До можливості оператора відносять перегляд та редагування особистих даних та відтворення скриптів.

1.2 Проектування конструктора скритів та його БД

Конструктор скриптів по праву можна вважати головною частиною веб-застосунку, отож до його проектування і розробки потрібно було підходити з особливою уважністю.

Конструктор має структуру що складається з трьох рівнів *(див рис 3.3)*.

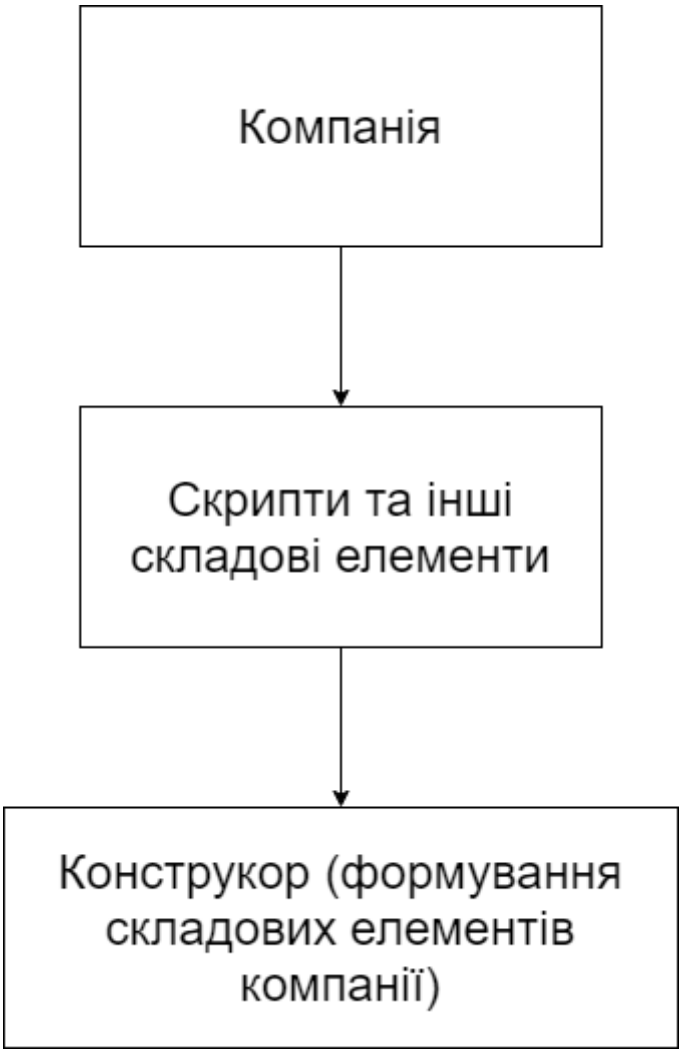


Рис 3.3 Структура скриптів та їх конструктора

1.2.1 Перший рівень конструктора

Компанія – складна сутність, яка в собі містить, сукупність розмовних скриптів, базу стандартних запитань та iframe веб-сторінки компанії *(див рис 3.4)*.

Рис 3.4 Схематичне зображення структури компанії

Тобто перший рівень являє собою таблицю компаній, з можливістю створення нових, та редагування і вилучення вже існуючих. На першому структурному рівні передбачений запуск компанії, що переносить нас безпосередньо до відображення сконструйованого другого рівня і запускає початковий скрипт розмови.

1.2.2 Другий рівень конструктора

Другий рівень являють собою складові компанії, головними складовими є скрипти розмов, як раніше зазначалось їх є три види, а саме: вхідний, вихідний та пропозиція. Також кожна компанія має свою базу стандартних запитань, які будуть відображатися при відтворенні скриптів під час розмови оператора.

1.2.3 Проектування вихідного скрипта, його структури

Схематичний образ вихідного скрипта це дерево, у більшості випадках бінарне. Де парні рівні вершин разом із початковою це репліки оператора або ж менеджера, а непарні відповідно клієнта *(див рис 3.5)*.

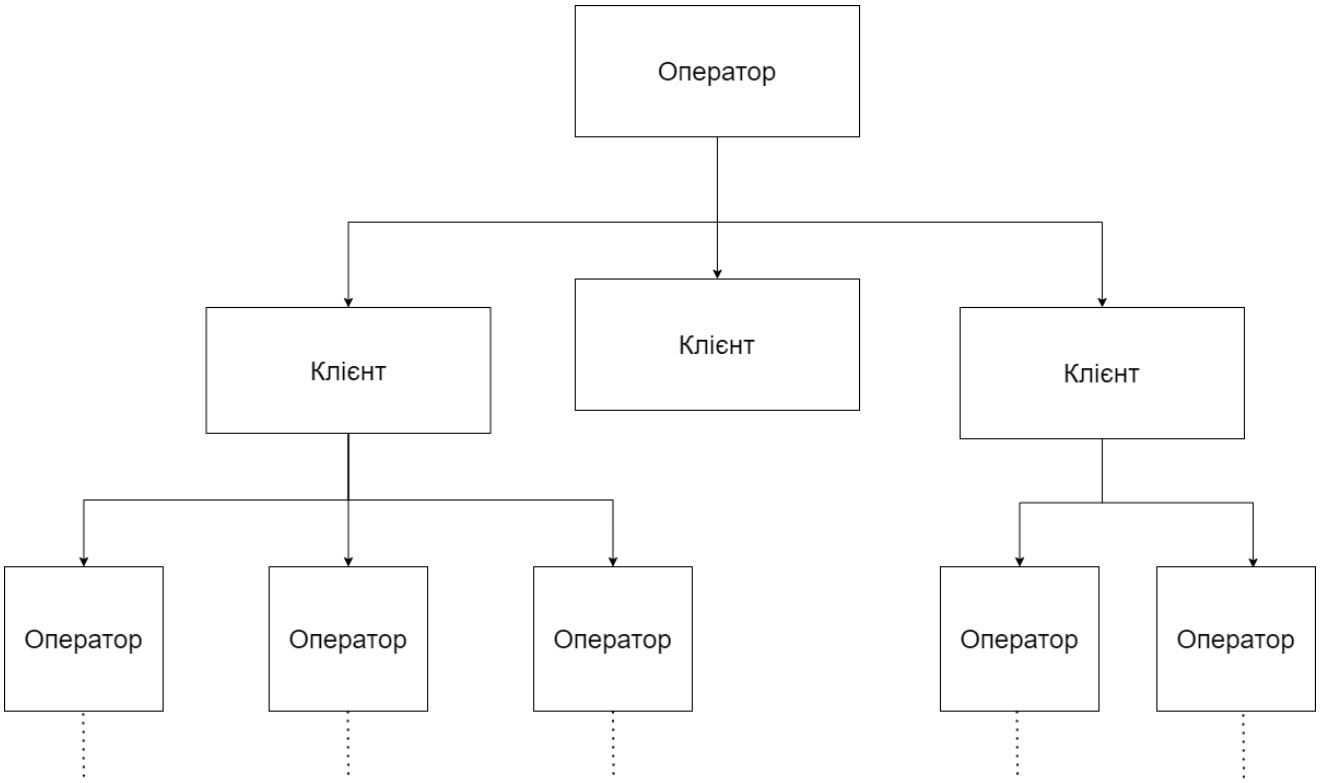


Рис 3.5 Схема сценарію розмови

Тобто вершини дерева це блоки скрипта. Всього існує 4 види блоків:

- Text (несе лише текстову інформацію)
- Button (кнопка що дозволяє рухатися далі по скрипту, містить репліку яку потрібно сказати, або ту яка очікується від клієнта)
- List (список варіантів відповідей або реплік)
- User field (поля що даються на вибір, з попередньо добавлених адміністратором, містять запит на інформацію про клієнта)

1.2.4 Вхідний скрипт

Має значно простішу структуру на відміну від вихідного. Вхідний скрипт являє собою анкету, його головна ціль отримати всі дані для оформлення замовлення. Тобто блоки залишилися ті ж що й у вихідному лише змінилася структура, тепер це просто послідовність блоків *(див рис 3.6)*.

Рис 3.6 Схема будови вхідного скрипта

1.2.5 Скрипт Пропозиція

Простіше кажучи це компонент програми, що дозволить створити лист який буде потрібно надіслати клієнту під час розмови, і для цього потрібно буде ввести лише адресу електронної пошти клієнта. Тобто це значно пришвидшує та спрощує процес надсилання потрібного електронного листа в потрібний для цього момент. Для реалізації даного функціоналу використовувалася технологія PHPmailer.

1.2.6 FAQ та iframe

Розділ FAQ являє собою базу стандартних питань, які при бажанні, можна відобразити паралельно відтворенню скрипта, відповідно для них створюється окрема таблиця яка буде містити ще два атрибута окрім ключа – це запитання та відповідь.

Iframe (Inline frame) – це веб-сторінка в середині веб-сторінки, дозволяє працівнику під час відтворення скрипта не переходити окремо по посиланню веб-сайту компанії, послуги якої вона пропонує, а зробити це безпосередньо з розробленого із застосунку, під час відтворення сценарію розмови. Таблиця даної сутності має лише два атрибута це назва та посилання на веб-сторінку.

1.2.7 Проектування БД

Зробивши детальний аналіз наведеної вище інформації можна зрозуміти що у нас є 6 видів сутностей:

- Компанія;
- Складові компанії (розмовні скрипти, FAQ, iframe);
- Блоки скрипта;

- User field.

Структуру зв’язків даної системи можна вважати середньої складності *(див рис 3.7)*

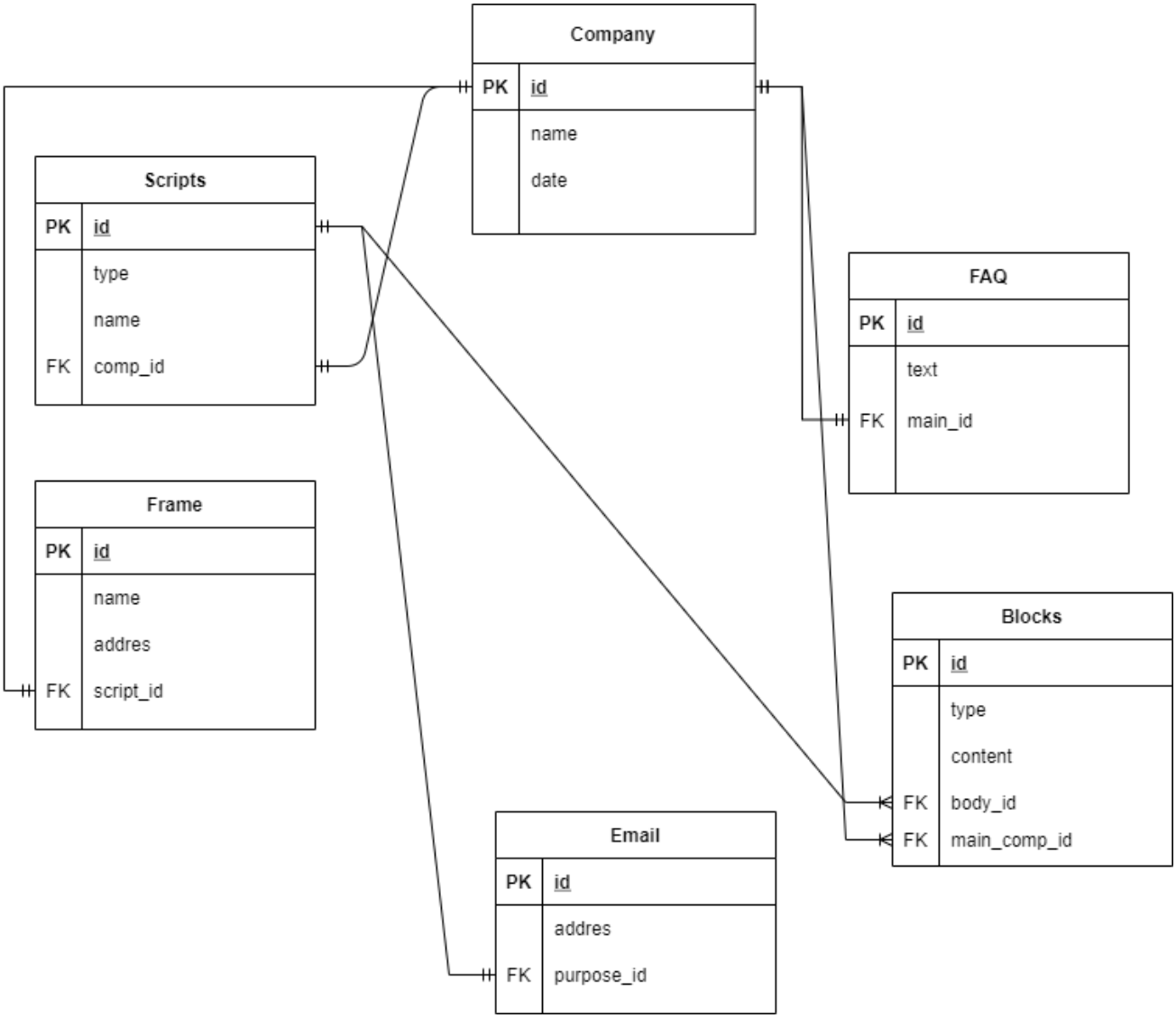


Рис 3.7 Схематичне зображення БД

Всього БД налічує 15 таблиць.

1.2.8 Таблиця Company

Таблиця Company має досить просту структуру містить лише два атрибути окрім ключа id яке має статус AUTO_INCREMENT, наступні атрибути name – назва компанії, date – дата створення *(див рис 3.8)*.

id	int(11)		Нет	*Нет*	AUTO_INCREMENT
name	text	utf8_unicode_ci	Нет		
date	date		Нет	*Нет*	

Рис 3.8 Структура Company

1.2.9 Таблиця Scripts

Таблиця містить в собі назви складових компаній їх тип та id компанії якій вони належать, налічує такі атрибути як, id ключовий, що має статус AUTO_INCREMENT, name – назва складової, type – статус сутності вибраний з такого переліку:

- Incoming;
- Outgoing;
- Purpose;
- FAQ;
- Iframe.

Також містить атрибут script_id, що позначає компанію до якої належить модуль розмови *(див рис 3.9)*.

id	int(11)		Нет *Нем*	AUTO_INCREMENT
name	text	utf8_unicode_ci	Нет	
type	text	utf8_unicode_ci	Нет	
script_id	int(11)		Нет *Нем*	

Рис 3.9 Структура Scripts

1.2.10 Таблиця Script blocks

Дана таблиця має дещо складнішу структуру за її попередників *(див рис 3.10)*. Містить наступні атрибути. Як і в попередніх таблицях наявне ключовий атрибут id із статусом AUTO_INCREMENT, priviousid позначає id попереднього блоку, значення атрибуту status набуває лише із такого списку:

- Text (текстова інформація);
- List (список);
- Button (репліка);
- User field.

Атрибут text заповнюється лише тоді коли status має значення “Text” або ж “Button”. Scriptid містить в собі id скрипта до якого належить відповідний блок, а останнє в списку listype визначає тип списку, в разі заповнення атрибута status значенням “List”.

id	int(11)		Нет *Нем*	AUTO_INCREMENT
priviousid	int(11)		Нет *Нем*	
status	text	utf8_unicode_ci	Нет	
text	text	utf8_unicode_ci	Нет	
scriptid	int(11)		Нет *Нем*	
listype	text	utf8_unicode_ci	Нет	

Рис 3.10 Структура Script blocks

1.2.11 Допоміжні таблиці

Всього існує три допоміжні таблиці окрім головних таблиць які наведені вище і відіграють ключову роль в конструюванні скриптів. Всі вони схожі за своєю структурою, тобто містять інформацію про відповідні сутності. Розглянемо таблицю Questions*(див рис 3.11)*.

id	int(11)		Нет	*Нет*	AUTO_INCREMENT
question	mediumtext	utf8_general_ci	Нет		
answer	mediumtext	utf8_general_ci	Нет		
id_script	int(11)		Нет	*Нет*	

Рис 3.11 Структура Questions

Таблиці User fields, Purpose, List мають схожу структуру, тобто в кожній з цих таблиць є атрибут script_id, що пов’язує кожне поле з відповідним скриптом, в деяких із цих таблиць також є такий атрибут як block_id який пов’язує, наприклад список і його складові з відповідним блоком скрипта *(див рис 3.12)*.

id	int(11)		Нет	*Нет*	AUTO_INCREMENT
email	varchar(1000)	utf8_general_ci	Нет	*Нет*	
text	mediumtext	utf8_general_ci	Нет		
note	mediumtext	utf8_general_ci	Нет		
script_id	int(11)		Нет	*Нет*	
name	varchar(1000)	utf8_general_ci	Нет	*Нет*	
topic	varchar(1000)	utf8_general_ci	Нет	*Нет*	
emailid	int(11)		Нет	*Нет*	

Рис 3.12 Структура Purpose

1.3 Головна БД та її проектування, синхронізація

На відміну від БД конструктора скриптів і всіх його складових, БД застосунку яка описує такі сутності як “Працівник”, “Тариф” та “Контакти”, є запозиченою, оскільки даний ПП синхронізований з моєю попередньою розробкою, тобто таблиці названих сутностей не потребували окремого проектування. На рисунку нижче ви можете ознайомитись із структурою таблиці Emploee *(див рис 3.13)*.

	#	Имя	Тип
☐	1	id	int(11)
☐	2	fitsrname	text
☐	3	surname	text
☐	4	phone_num	varchar(30)
☐	5	email	text
☐	6	date	date
☐	7	address	mediumtext

Рис 3.13 Структура Emploee

Головна ціль синхронізації полягає в моніторингу роботи працівників, і спільній статистиці, відповідно для цього база працівників та розкладу роботи спільна.

1.4 Програмна розробка веб-застосунку та її складові частини

Спираючись на всі проведені дослідження та на спроектовану структуру проекту разом з БД, наступним кроком постає програмна розробка у відповідному порядку *(див рис 3.14)*.

1.4.1 Frontend та дизайн

Перед розробкою дизайну головним питанням постав вибір головного діапазону кольорів. Головна схема кольорів *(див рис 3.15)*.

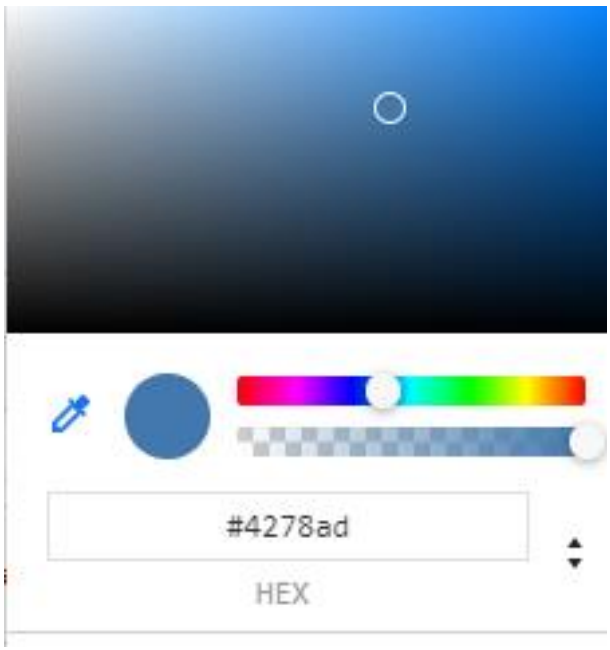


Рис 3.15 Схема кольорів

Головним завданням в розробці дизайну, стало створення інтуїтивно зрозумілої, зручної у використанні та візуально приємної веб-системи. Тому всі дії були спрямовані на те щоб максимально оптимізувати роботу програми завдяки правильному розташуванню робочих елементів див *(рис 3.16)*.

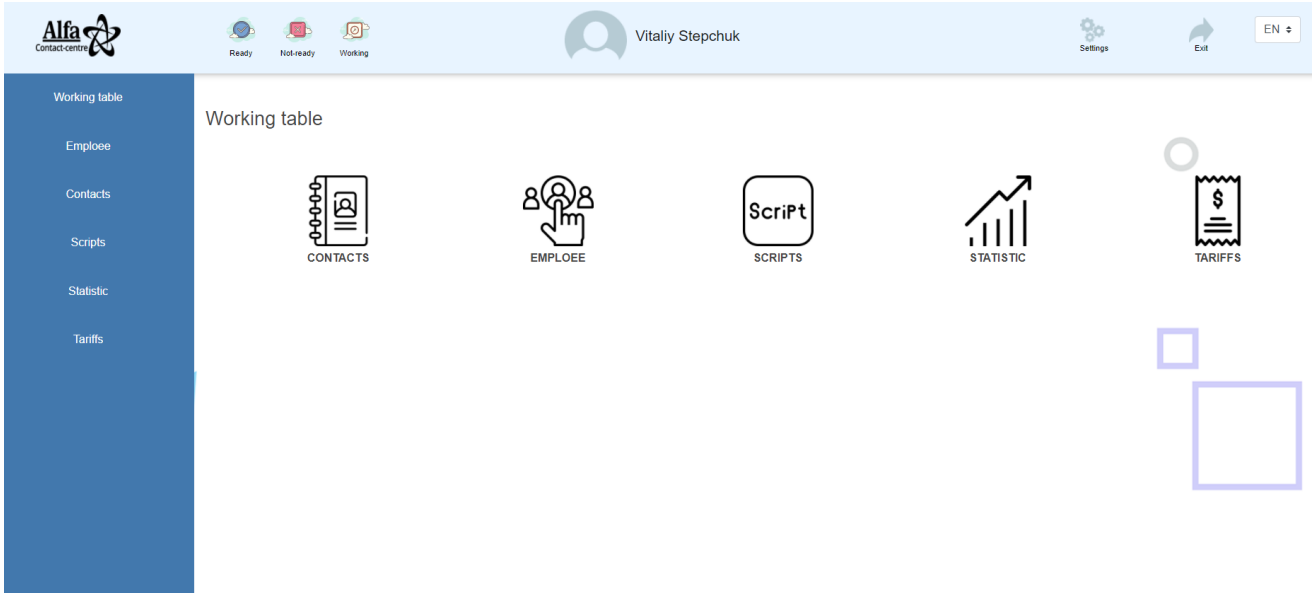


Рис 3.16 Скриншот головної сторінки

Як раніше згадувалося Frontend складова була розроблена за допомогою html5 та css3, завдяки CSS Flexible Box Layout Module не виникло жодних проблем із css розміткою, а саме порядком і вирівнювання елементів по декількох осях, розподіл вільного місця між елементами та іншими поширеними проблемами

1.4.2 Backend

Головною мовою Backend розробки, як раніше зазначалося стала мова програмування PHP а головним середовищем PHP Storm. Окрім вище згаданої мови програмування, також застосовувався JavaScript, а саме Jquery - у відправці даних з подальшим записом в БД, плагін tablesorter, що використовувався для реалізації динамічного редагування деяких таблиць за допомогою ajax запитів, та посторінкової навігації. Також було використано технологію PHP mailer та технологію AMI для взаємодії з Asterisk. При цьому більшість функціоналу ПП відбувається за допомогою POST методів.

1.4.3 Форма авторизації та її огляд

Форма авторизації передбачає два способи входу. Адміністратора пароль якого створює і редагує сам адміністратор при першій авторизації, та працівника, тобто оператора, який може авторизуватися у веб-застосунку увівши лише свій унікальний код, який присвоюється кожному працівнику, котрий реєструється в попередньо розробленому застосунку для систематизації розкладу роботи, що в свою чергу синхронізований з даним проектом. Форма авторизації користувачів має достатньо простий вигляд *(див рис 3.17)*. Для безпеки паролів була вибрана хеш-функція password_hash().

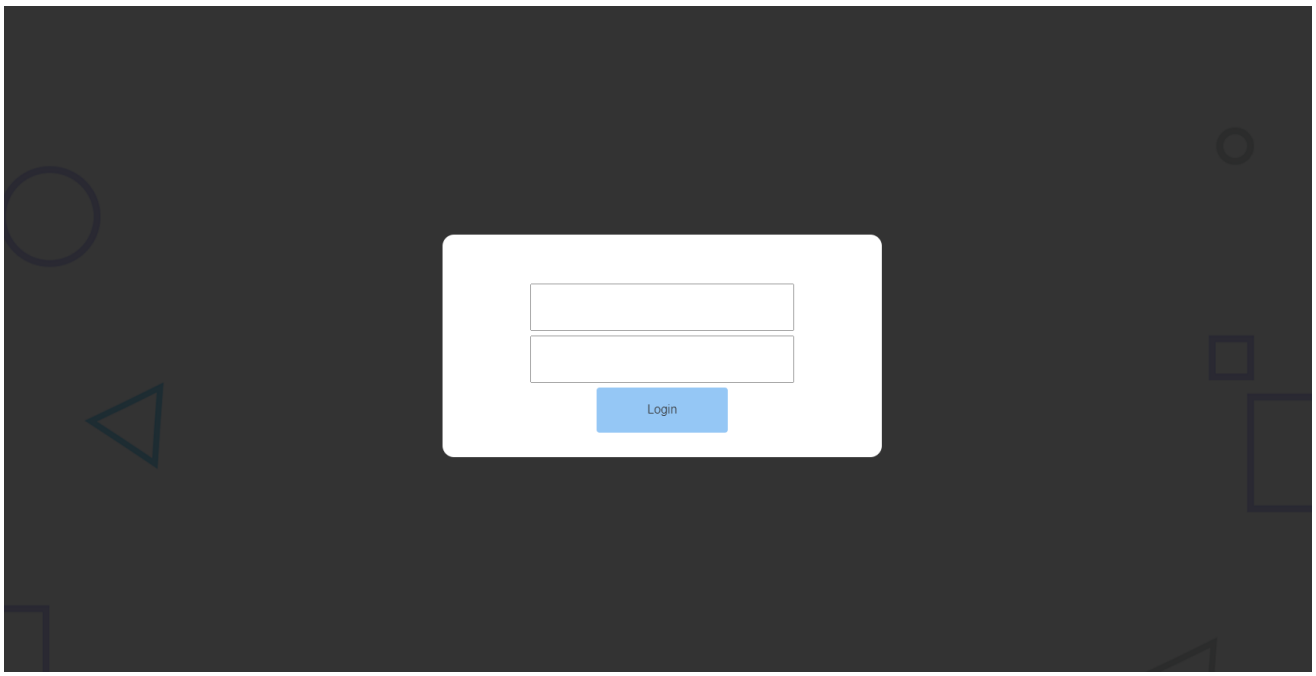


Рис 3.17 Форма авторизації користувачів

1.4.4 Головна сторінка, головне меню та їх огляд

Головна сторінка має навігаційний характер, тобто після авторизації кожен користувач потрапляє на головну сторінку, де може обрати подальший шлях. Вона

як і більшість сторінок містить закріплений header та головне меню *(див. рис. 3.18).*

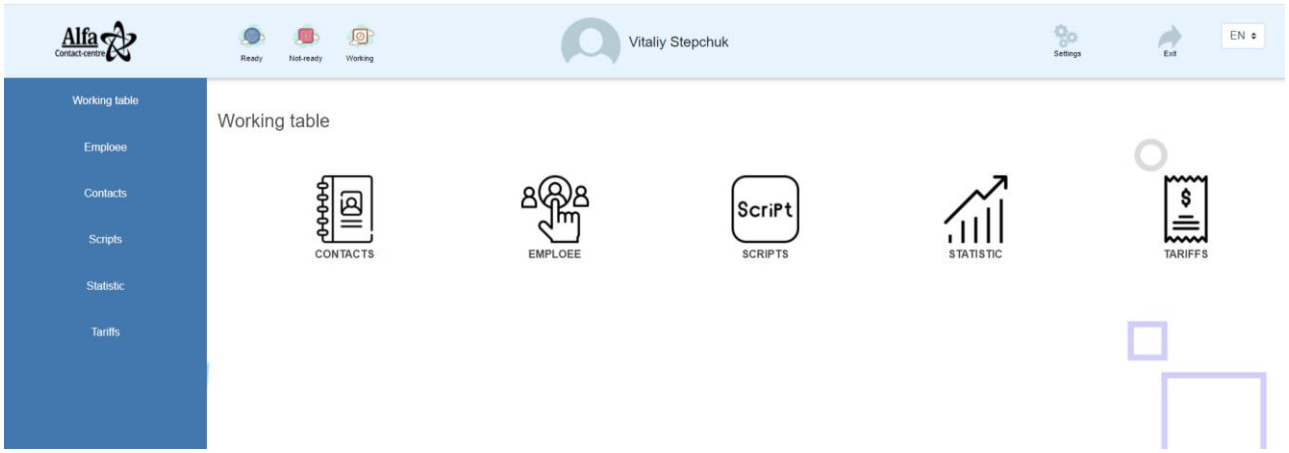


Рис. 3.18 Головна сторінка

Header містить в собі такі елементи:

- Логотип компанії
- Модуль активності користувача
- Особисту сторінку користувача
- Налаштування
- Модуль що відповідає за розлогін
- Зміну головної мови веб-сервісу

Даний елемент прописаний в окрему файлі *(див рис 3.19)* і є підключеним до кожної сторінки веб ресурсу, окрім відтворення скриптів.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light main_header_block">
    <nav class="navbar navbar-light bg-light logo">
        <a class="navbar-brand" href="http://crm.ccalfa.com/main.php">
            <img src="images/mainlogo.png" width=120 height="60" alt="">
        </a>
    </nav>

    <div class="row user_status">

        <div class="col-lg-4 col-md-4 col-xs-4 thumb flex justify-content-center">
            <a class="thumbnail" href="#">
                <img src="images/ready.png" width="50" height="50" alt="">
                <p>Ready</p>
            </a>
        </div>
        <div class="col-lg-4 col-md-4 col-xs-4 thumb flex justify-content-center">
            <a class="thumbnail" href="#">
                <img src="images/not_ready.png" width="50" height="50" alt="">
                <p>Not-ready</p>
            </a>
        </div>
        <div class="col-lg-4 col-md-4 col-xs-4 thumb flex justify-content-center">
            <a class="thumbnail" href="#">
                <img src="images/working.png" width="50" height="50" alt="">
                <p>Working</p>
            </a>
        </div>
    </div>
    <nav class="navbar navbar-light bg-light person">
        <a class="navbar-brand person_photo" href="loginperson.php">
            <img src="images/noavatar.png" width="100" height="70" alt="">
```

Рис 3.19 Частина коду хедера

1.4.5 Загальний вигляд таблиць та їх функціонал

Всі таблиці що є у веб застосунку мають один дизайн и схожий функціонал. Розглянемо для прикладу таблицю Conctacts, в яку записуються постійні клієнти компанії *(див рис 3.20)*.

Contacts

First name	Last name	Company	Date of next step	Phone num.	Manager	Result	
adas	asd	asdasd	2020-09-02	311	31	1	
eee	ddd	dasd	0000-00-00	380657849499	3	1	

Рис 3.20 Таблиця Contats

При роботі з кожною таблицею ми маємо ряд стандартних можливостей:

- Додавання нових записів
- Редагування
- Видалення

Редагування і додавання мають схожий вигляд *(див рис 3.21)* і функціонал, реалізовані вони наступним чином, кожна інформаційна сторінка містить свій “Engine” тобто файл, у якому обробляються всі запити, що і реалізовують наявний функціонал сторінки. Після успішного редагування, або ж відміни, користувач повертається до сторінки де зображено таблицю.

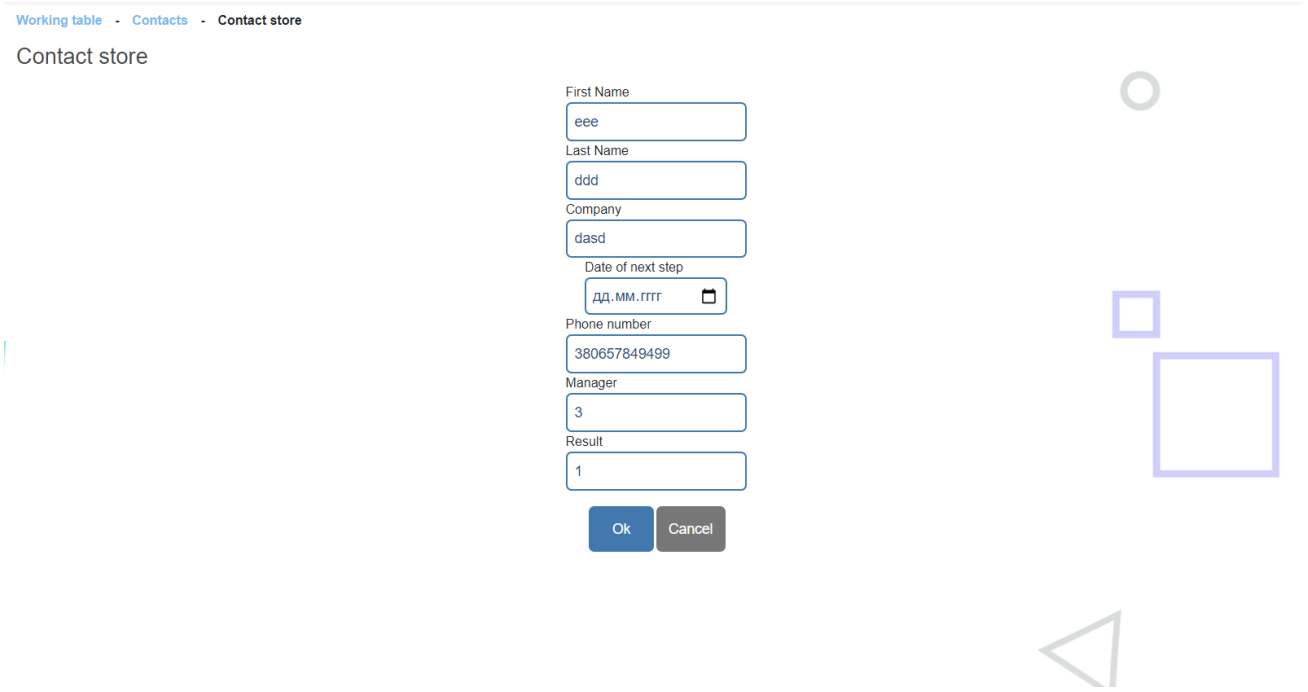


Рис 3.21 Форма редагування даних

1.4.6 Скрипти розмов та їх конструктори

Одною із найголовніших частин програми являються саме скрипти та модулі що взаємодіють з ними. Для прикладу розглянемо найскладніший вид скрипта та його конструктор, коли користувач натискає в головному меню на вкладку Scripts він потрапляє на сторінку де зображено таблицю *(див рис 3.22),* в якій

відображаються всі створені компанії. Після натискання на значок редагування, користувача переносить на сторінку де відображені всі структурні елементи компанії, з можливістю додавати нові та редагувати і усувати існуючі.

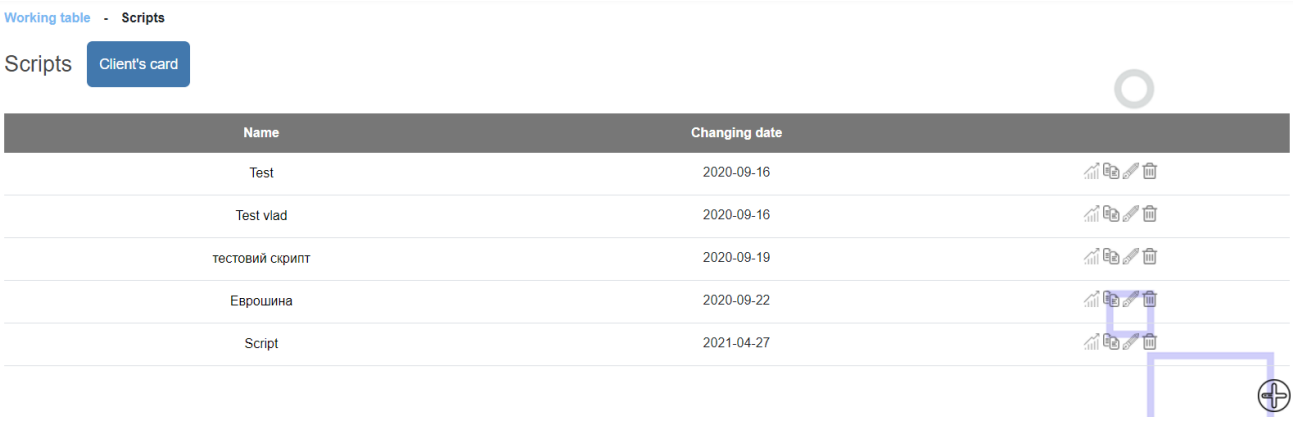


Рис 3.22 Таблиця компаній

Вигляд таблиці структурних елементів скрипта є ідентичним стандартному вигляду таблиць, після натискання на іконку редагування, користувача перенаправляє на відповідний конструктор в залежності від типу елемента.

Конструктор вихідного скрипта має вигляд робочого поля *(див рис 3.23)*, в якому у відповідній послідовності відображені його складові блоки. Кожен блок має відповідне призначення. Під час додавання нового блоку, користувач вибирає його тип а потім і його наповнення. Також на кожному структурному елементі конструктора відображається іконка, що відображає особу якій належить даний елемент, всього є лише два варіанти: оператор, клієнт. Також кожен блок містить три кнопки: редагування, видалення та додавання наступного блоку. Для прикладу видалення блоків було реалізовано за допомогою рекурсивної функції *(див рис 3.24)*. Тип елемента відображається в текстовому форматі у лівому нижньому куті.

Рис 3.23 Конструктор вихідного розмовного скрипта

```php
function delBlock($id,$mysqli) { //Рекусивная функцыя удаления всех блоков которые следуют за основным
    $result = $mysqli->query("SELECT * FROM `Scripts` WHERE `priviousid`='$id'");
    while($row = $result->fetch_assoc())

    if(isset($row['id'])){
       $idd=$row['id'];
       echo $idd;
       delBlock($idd,$mysqli);
       $mysqli->query("DELETE FROM `Scripts` WHERE `id`='$idd'");
    }
    else{
       return $mysqli->query("DELETE FROM `Scripts` WHERE `id`='$id'");
    }
}
```

Рис 3.24 Функція рекурсивного видалення блоків

Перший тип це Button – його головна функція це зображення репліки яку потрібно озвучити оператору або ж репліки клієнта *(див рис 3.25)*, що очікується у відповідь, тобто даний тип блоків є найбільш використовуваним оскільки

відіграє роль послідовного відтворення сценарію розмови. Наповнюється текстом.

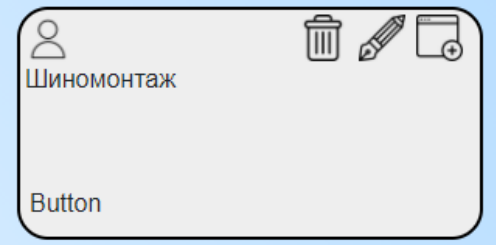
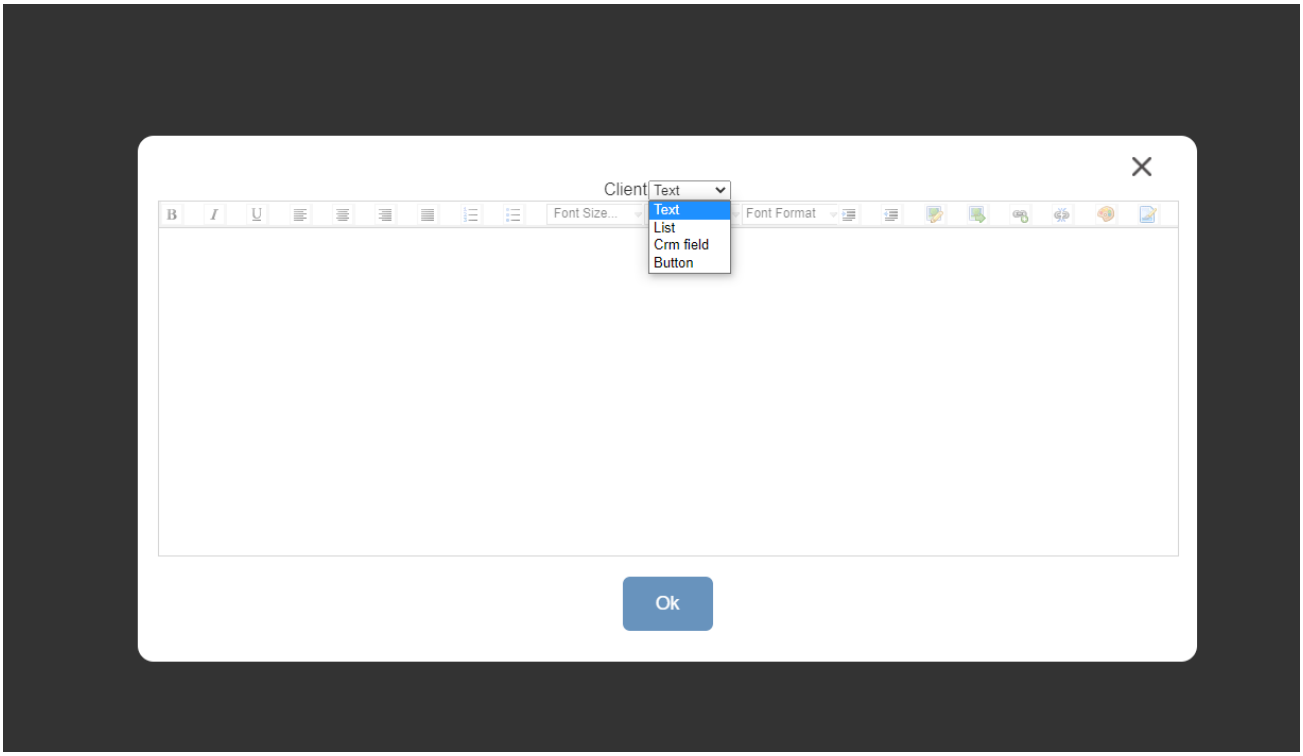


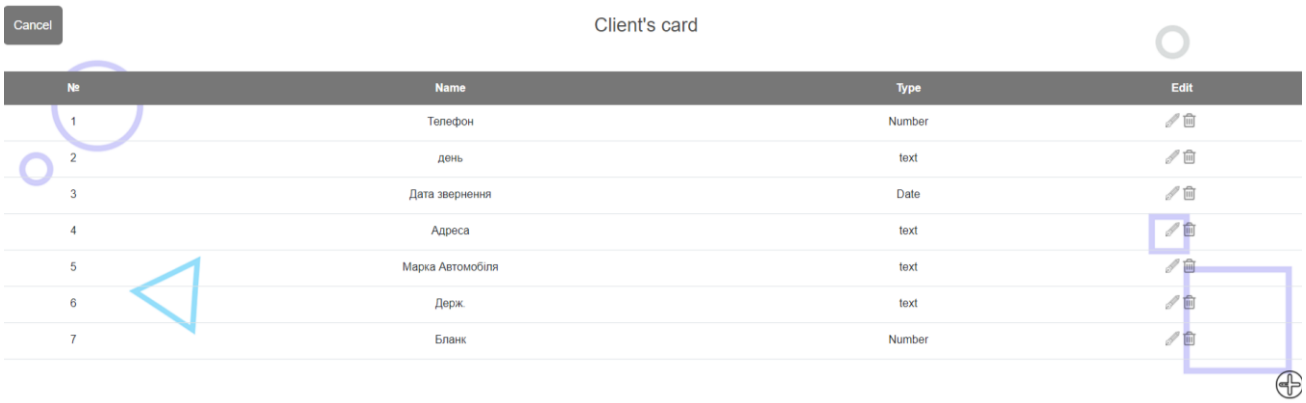
Рис 3.25 Конструктор вихідного розмовного скрипта

Другий тип який використовується досить часто і є не менш важливим за попередній це List – блок що відповідає за створення списку під час відтворення скрипта, в ту ж чергу він поділяється на три типи: моноваріантний, багатоваріантний та випадаючий список. Під час додавання даного блоку у користувача також є можливість вибору варіанту відображення списку: рядок, стовпчик та шаховий варіант *(див рис 3.26)*.



3.26 Форма додавання блоку скриптає

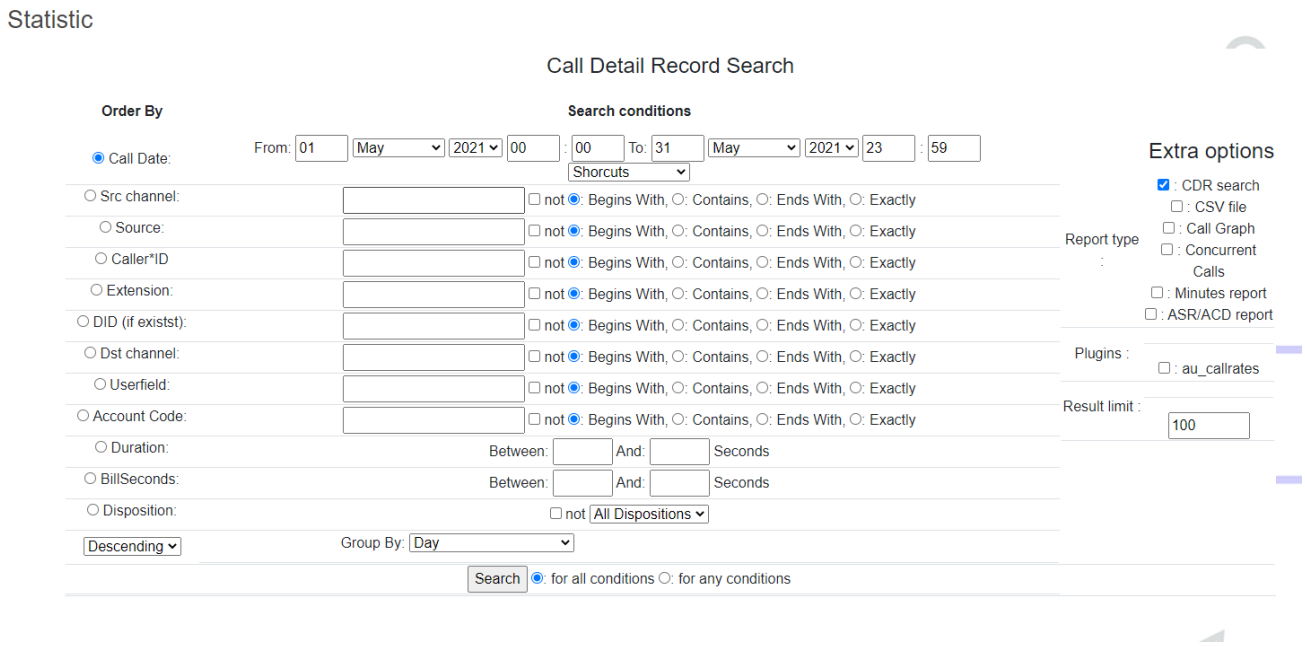
Наступний тип – це стандартні інформаційні поля клієнта, що додаються менеджером на окремій сторінці, на яку можна потрапити із сторінки компаній. Такими полями для прикладу являються мобільний телефон, адреса, дата звернення, тощо. Даний блок призначений для спрощеного конструювання сценарію розмови та ефективнішої роботи оператора під час його відтворення *(див рис 3.27)*.



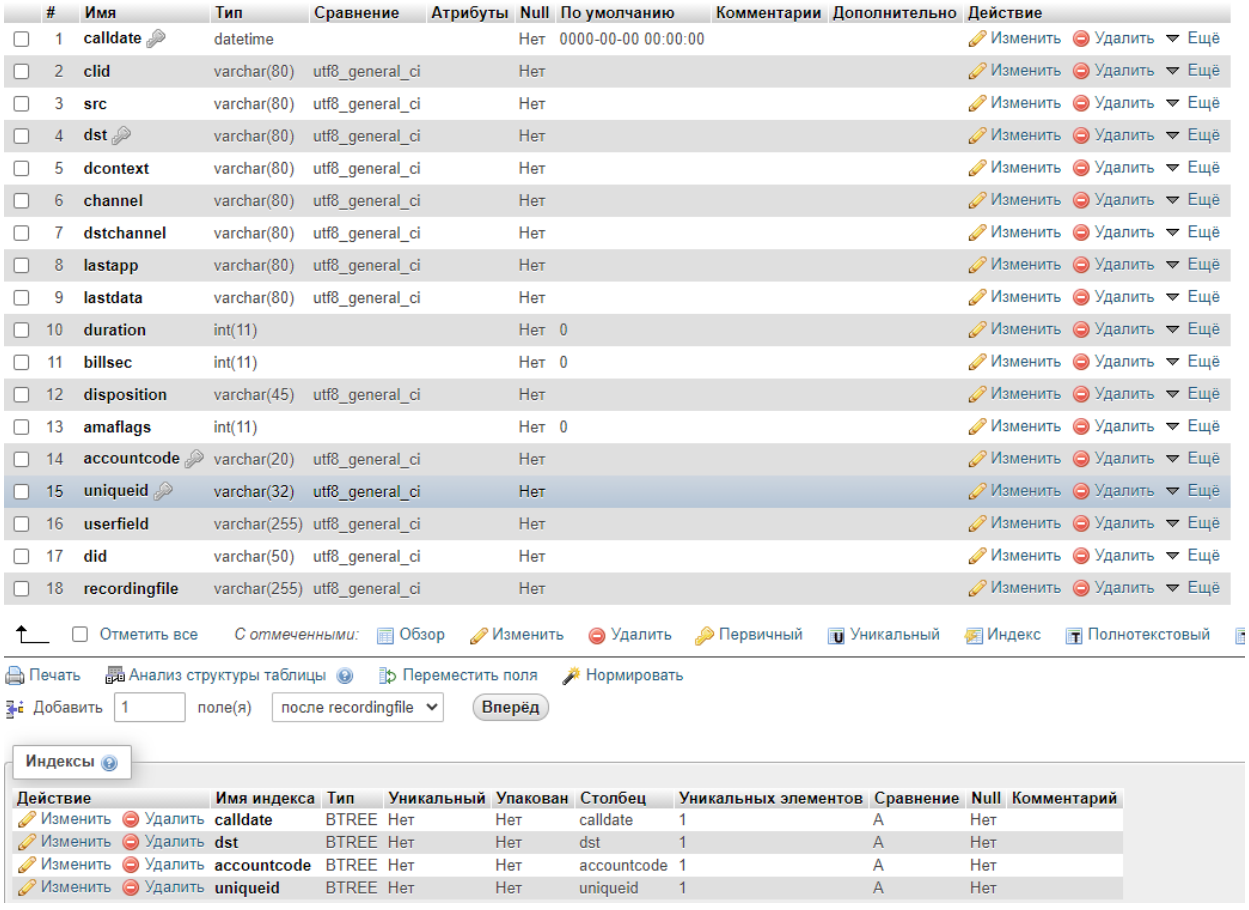
3.27 Таблиця стандартних полів клієнта

1.4.7 Статистика

Головною метою статистики є легко зрозуміле і максимально широке відображення всіх даних що циркулюють веб застосунком, для відображення статистики окрім власноруч написаних файлів, також використовувався готовий модуль “cdr” *(див рис 3.28)* статистики дзвінків, та готова таблиця даної статистики *(див рис 3.29)*.



3.27 Форма запиту статистики дзвінків



3.28 Структура таблиці статистики

Таблиця статистики даного модулю заповнюється автоматично системою Asterisk, а саме за допомогою технології-взаємодії AMI.

1.4.8 Статус активності та його моніторинг

Однією із головних функцій веб застосунку, являється моніторинг активності користувачів *(див рис 3.29)*.

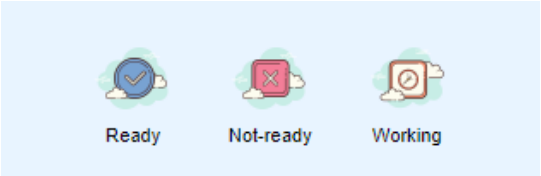


Рис 3.29 Блок що відповідає за активність користувачів

Як тільки користувач авторизується на сайті йому автоматично присвоюється статус “Ready” і відповідно робиться запис в базу даних *(див рис 3.30)*, коли відповідний користувач авторизувався. Окрім статусу “Ready” є ще такий статус як “Not-ready” і “Working”, кожна активізація будь-якого із статусів записується в БД, для подальшого моніторингу з боку менеджера.

☐	Изменить	Копировать	Удалить	222	7	last_activity	11:05:47	2020-11-09
☐	Изменить	Копировать	Удалить	223	7	exit	11:05:49	2020-11-09
☐	Изменить	Копировать	Удалить	224	1	last_activity	11:05:53	2020-11-09
☐	Изменить	Копировать	Удалить	225	1	last_activity	08:59:53	2020-11-10

Рис 3.30 Записи в БД

Якщо ж користувач є неактивним протягом 10 хв, то в БД записується час його останньої активності та відбувається автоматичний розлогін, далі користувач буде змушений пояснити причину своєї бездіяльності.

1.5 Огляд автоматизації системи

Головним рушієм автоматизації системи, стала вище згадана технологія AMI [6].

Було попередньо налаштовано Asterisk. Далі у окремому файлі було прописано функції взаємодії з сервером. Головною є функція що дозволяє ініціалізувати виклик і в подальшому визначити його деталі, що дозволять системі визначити який скрипт повинен бути запущеним *(див рис 3.31)*.

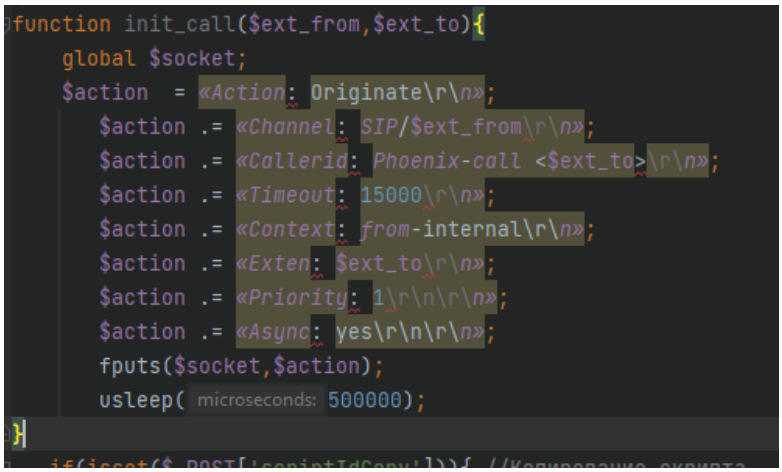


Рис 3.31 Функція ініціалізації виклику

Оператор отримує дзвінок на свій телефон, далі провіряє останній виклик за допомогою модулю статистики “cdr” де буде відображено назву компанії що телефонує, тоді оператор розуміє який скрипт розмови повинен бути запущений.

Залишається лише чітко прослідувати по заготовленому сценарію і успішно завершити розмову з клієнтом.

ВИСНОВКИ

У даній роботі був проведений детальний аналіз роботи call-центрів, та всіх його складових, було приділено особливу увагу аналізу сценаріїв розмов операторів call -центру та клієнтів. На основі здобутих даних було розроблено веб-застосунок що значно збільшив ефективність, якість та швидкість роботи працівників під час розмов.

Проведено порівняльний аналіз існуючих на ринку технологій з автоматизації роботи call-центрів, виділено переваги та недоліки кожного з них, розглянуто архітектуру конструкторів розмовних скриптів. Проведено аналіз проблем, які виникають під час розробки аналогічних програмних застосунків. В результаті було спроектовано гнучкий конструктор скриптів та автоматизовано всі процеси взаємодіє операторів та клієнтів за допомогою технології AMI.

Спроектовано та розроблено веб-систему з використанням сучасних технологій, що переважає у своїй ефективності над уже існуючими системами. Також розроблений веб-застосунок був протестований call-центром і використовується в реальному часі.

Отже можна зробити головний висновок, що актуальність проблеми автоматизації роботи операторів call-центру є дуже високою, можна і на далі покращувати розроблену систему, робити її більш досконалою у використанні при цьому збільшуючи швидкість роботи та ефективність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. AMI technology in php [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://voxlink.ru/kb/linux/ami-funkcional-iz-php/
2. PHP manual [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.php.net/manual/ru/index.php
3. Practical Asterisk 1.4 and 1.6: From Beginner to Expert (First Edition)
4. AMI interface of management Asterisk [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://voxlink.ru/kb/book/interfejs-upravlenija-asterisk-ami/
5. Call Center: A Focus on Customer Service (First Edition)
6. Asterisk php manager [Електронний ресурс]. – 2020. – https://github.com/OdinsHat/asterisk-php-manager